

# **Introduction to Big Data Assignment-4**

**Name:** Faizan Mulla

**Roll No:** 21F1003885

## **Problem Statement**

Write PySpark code to implement SCD Type II on a customer master data frame. You will have to create the input files by yourself. For this, use the input data shown as the example in the lecture when this topic was discussed. The PySpark code should be executed on a Dataproc cluster.

Note that within PySpark, there are multiple ways to solve this problem. Any of the ways is acceptable. But do NOT use SparkSQL.

## **Solution**

### ***Environment Setup***

#### 1. GCP Setup

- Set up a new project with default service account configurations
- Enabled necessary APIs (Compute Engine and Dataproc)

#### 2. Dataproc Cluster Creation

- Created a Dataproc cluster with following specifications:
  - chose the option: Create cluster on compute engine
  - manager node: series → e2 // machine type → e2-standard-2 (2vCPU, 1 core, 8GB memory)
  - reduce primary disk size from 500GB to something less like 50GB.
  - exact same settings for worker nodes too.
- Region: **asia-south-1**
- in the customize cluster menu, **uncheck** the INTERNAL IP ONLY option.

### 3. Cloud Storage Setup

- Created a Cloud Storage bucket for storing the original and the updated CSV files + python file
- Configured with standard storage class
- Region: **asia-south-1**

### Implementation Details

```
SCD-2.py X
1  from datetime import datetime
2  from pyspark.sql import SparkSession, functions as F
3  from pyspark.sql.functions import when
4
5
6  current_date = datetime.now().strftime("%d-%m-%Y")
7
8  # Create a spark session
9  spark = SparkSession.builder.appName("SCD_Type_2").getOrCreate()
10
11 # Access data
12 original = spark.read.csv("gs://iitm-ibd-ga4/original_data.csv", header=True, inferSchema=True)
13 updated = spark.read.csv("gs://iitm-ibd-ga4/updated_data.csv", header=True, inferSchema=True)
14
15 for row in updated.collect():
16     condition1 = F.col("name") == row.name
17     condition2 = F.col("end_date") > current_date
18
19     original = original.withColumn(
20         "end_date",
21         when(condition1 & condition2, current_date).otherwise(original.end_date),
22     )
23
24 for row in updated.collect():
25     idx_ = original.tail(1)[0].idx + 1
26     name_ = row.name
27     dob_ = row.dob
28     tuple_ = (idx_, name_, dob_, current_date, "10-12-2099")
29
30     row_ = spark.createDataFrame([tuple_], schema=original.schema)
31     original = original.union(row_)
32
33 # Show the data
34 original.show()
35
36 # Save to CSV and stop the spark session
37 original.toPandas().to_csv("output.csv", index=False, header=True)
38 spark.stop()
```

## Execution Process

### 1. Data Generation / Uploading

- Uploaded ***original\_data.csv*** & ***updated\_data.csv*** files to Cloud Storage Bucket.
- Changed the file names in the Python script to the gsutil URI of the respective files.

### 2. SCD-2

- Now, upload ***SCD-2.py*** to Cloud Storage
- Created Dataproc Cluster
- Submitted Spark job through Dataproc
- Monitored job execution through Dataproc UI

## Results

≡ output.txt ✕	
1	+---+-----+-----+-----+-----+
2	idx     name         dob start_date   end_date
3	+---+-----+-----+-----+-----+
4	1   Alice 1990-01-01 01-01-2023 04-11-2024
5	2     Bob 1985-05-15 01-02-2023 10-12-2099
6	3 Charlie 1992-09-09 01-03-2023 10-12-2099
7	4     Eve 1987-12-12 01-04-2023 04-11-2024
8	5   Frank 1993-06-22 01-05-2023 10-12-2099
9	6   Alice 1990-01-01 04-11-2024 10-12-2099
10	7   David 1988-08-08 04-11-2024 10-12-2099
11	8     Eve 1987-12-12 04-11-2024 10-12-2099
12	9   Grace 1991-10-10 04-11-2024 10-12-2099
13	10   Henry 1995-03-03 04-11-2024 10-12-2099
14	+---+-----+-----+-----+-----+

Relevant Screenshots

1. Cloud Storage Bucket Contents

iitm-ibd-ga4

Location

asia-south1 (Mumbai)

Storage class

Standard

Public access

Not public

Protection

Soft Delete

OBJECTS

CONFIGURATION

PERMISSIONS

PROTECTION

LIFECYCLE

OBSERVABILITY

INVENTORY REPORTS

OPERATIONS

Folder browser

iitm-ibd-ga4

Buckets > iitm-ibd-ga4

CREATE FOLDER

UPLOAD

TRANSFER DATA

OTHER SERVICES

Filter by name prefix only

Filter

Filter objects and folders

Show Live objects only

<input type="checkbox"/>	Name	Size	Type	Created	Storage class	
<input type="checkbox"/>	SCD-2.py	1.2 KB	application/octet-stream	Nov 4, 2024, 12:45:33 PM	Standard	<div><div>Download</div><div>More</div></div>
<input type="checkbox"/>	original_data.csv	242 B	text/csv	Nov 4, 2024, 12:44:36 PM	Standard	<div><div>Download</div><div>More</div></div>
<input type="checkbox"/>	updated_data.csv	98 B	text/csv	Nov 4, 2024, 12:44:33 PM	Standard	<div><div>Download</div><div>More</div></div>

2. Dataproc Cluster Configuration

Cluster details

SUBMIT JOB

REFRESH

START

STOP

DELETE

VIEW LOGS

Consider using Auto Zone rather than selecting a zone manually. See https://cloud.google.com/dataproc/docs/concepts/configuring-clusters/auto-zone

Name	iitm-ibd-ga4
Cluster UUID	88c62af7-9ff5-45ab-9636-fa0cb06f92ae
Type	Dataproc Cluster
Status	<div>Running</div>

3. Job Execution Results

Job details

CLONEDELETESTOPREFRESH

Job IDiitm-ibd-ga4-job

Job UUIDfef8d047-a8c8-4150-8c1c-e8da95b121f2

TypeDataproc Job

StatusSucceeded

MONITORING

CONFIGURATION

Output

LINE WRAP: OFF

24/11/04 07:23:44 INFO GoogleCloudStorageImpl: Ignoring exception of type GoogleJsonResponseException; verified object already exists with desired state.

24/11/04 07:23:45 INFO GoogleHadoopOutputStream: hflush(): No-op due to rate limit (RateLimiter[stableRate=0.2qps]): readers will \*not\* yet see flushed data for gs://dataproc-temp-asia-sc

idx	name	dob	start_date	end_date
1	Alice	1990-01-01	01-01-2023	04-11-2024
2	Bob	1985-05-15	01-02-2023	10-12-2099
3	Charlie	1992-09-09	01-03-2023	10-12-2099
4	Eve	1987-12-12	01-04-2023	04-11-2024
5	Frank	1993-06-22	01-05-2023	10-12-2099
6	Alice	1990-01-01	04-11-2024	10-12-2099
7	David	1988-08-08	04-11-2024	10-12-2099
8	Eve	1987-12-12	04-11-2024	10-12-2099
9	Grace	1991-10-10	04-11-2024	10-12-2099
10	Henry	1995-03-03	04-11-2024	10-12-2099

24/11/04 07:24:30 INFO DataprocSparkPlugin: Shutting down driver plugin. metrics=[action\_http\_patch\_request=0, files\_created=1, gcs\_api\_server\_timeout\_count=0, op\_get\_list\_status\_result\_s

Job iitm-ibd-ga4-job successfully submitted

4. Output File Contents

Output

LINE WRAP: OFF

24/11/04 07:23:33 INFO SparkEnv: Registering MapOutputTracker

24/11/04 07:23:33 INFO SparkEnv: Registering BlockManagerMaster

24/11/04 07:23:33 INFO SparkEnv: Registering BlockManagerMasterHeartbeat

24/11/04 07:23:33 INFO SparkEnv: Registering OutputCommitCoordinator

24/11/04 07:23:35 INFO MetricsConfig: Loaded properties from hadoop-metrics2.properties

24/11/04 07:23:35 INFO MetricsSystemImpl: Scheduled Metric snapshot period at 10 second(s).

24/11/04 07:23:35 INFO MetricsSystemImpl: google-hadoop-file-system metrics system started

24/11/04 07:23:36 INFO DataprocSparkPlugin: Registered 188 driver metrics

24/11/04 07:23:37 INFO DefaultNoHARMFalloverProxyProvider: Connecting to ResourceManager at iitm-ibd-ga4-m.asia-south1-a.c.intro-to-big-data-439410.internal./10.160.0.5:8032

24/11/04 07:23:37 INFO AHSProxy: Connecting to Application History server at iitm-ibd-ga4-m.asia-south1-a.c.intro-to-big-data-439410.internal./10.160.0.5:10200

24/11/04 07:23:38 INFO Configuration: resource-types.xml not found

24/11/04 07:23:38 INFO ResourceUtils: Unable to find 'resource-types.xml'.

24/11/04 07:23:40 INFO YarnClientImpl: Submitted application application\_1738704738228\_0001

24/11/04 07:23:42 INFO DefaultNoHARMFalloverProxyProvider: Connecting to ResourceManager at iitm-ibd-ga4-m.asia-south1-a.c.intro-to-big-data-439410.internal./10.160.0.5:8030

24/11/04 07:23:44 INFO RequestTracker: Detected high latency for [url=https://storage.googleapis.com/storage/v1/b/dataproc-temp-asia-south1-671330585240-ygsijyny/o?delimiter=/&fields=item

24/11/04 07:23:44 INFO GHfsGlobalStorageStatistics: periodic connector metrics: [gcs\_api\_client\_non\_found\_response\_count=1, gcs\_api\_client\_side\_error\_count=1, gcs\_api\_time=859, gcs\_api\_to

24/11/04 07:23:44 INFO GoogleCloudStorageImpl: Ignoring exception of type GoogleJsonResponseException; verified object already exists with desired state.

24/11/04 07:23:45 INFO GoogleHadoopOutputStream: hflush(): No-op due to rate limit (RateLimiter[stableRate=0.2qps]): readers will \*not\* yet see flushed data for gs://dataproc-temp-asia-so

idx	name	dob	start_date	end_date
1	Alice	1990-01-01	01-01-2023	04-11-2024
2	Bob	1985-05-15	01-02-2023	10-12-2099
3	Charlie	1992-09-09	01-03-2023	10-12-2099
4	Eve	1987-12-12	01-04-2023	04-11-2024
5	Frank	1993-06-22	01-05-2023	10-12-2099
6	Alice	1990-01-01	04-11-2024	10-12-2099
7	David	1988-08-08	04-11-2024	10-12-2099
8	Eve	1987-12-12	04-11-2024	10-12-2099
9	Grace	1991-10-10	04-11-2024	10-12-2099
10	Henry	1995-03-03	04-11-2024	10-12-2099